Monocular depth estimation from single image

Piyush Gulve
Stanford
pgulve@stanford.edu

Gaurav Gupta
Stanford
guptag@stanford.edu

Abstract

In this project, we tackle the problem of depth estimation from a single image. We reviewed several different techniques for depth estimation from monocular images.

The first approach builds on top of a fully convolutional network upsampling part of the network [8]. The second approach uses encoder-decoder architecture using transfer learning, the encoder uses RGB images encoded into a feature vector using ResNet 34 [[1]]. Both the techniques have been proven to be better than fully connected convolutional neural networks which we did not pursue in this project.

1. Introduction

Depth information in computer vision has applications in various fields, including SLAM, AR and VR applications, object detection, semantic segmentation, etc. Depth estimation has been a topic of research for over 40 years including interpreting depth from geometric properties of image such as Structure from Motion (SFM) using a series of 2D Images or Stereo Vision Matching recovers 3D structures from observing the scene from two viewpoints, simulating how human eyes have two cameras. Other techniques, such as Feature-based mapping methods, rely on the assumption that similarities between regions in the RGB images also imply similar depth cues Sensor-based methods utilize depth sensors, like RGB-D cameras and Lidar. Finally, Deep Learning, various neural network architectures have been used to predict depth estimation.

Depth estimation from single or monocular images often arises in practice, such as better understanding of the many images distributed on the web and social media outlets, real estate listing, etc, which include both indoor and outdoor examples. Depth estimation also improves many other computer vision tasks when compared to RGB only methods such as in object recognition and semantic segmentation. In practice it is easier to get more data for monocular images, as compared to stereo or continuous stream of image or videos of place.

Our project will evaluate the CNN architectures for monocular depth estimation. Future work can be done to implement State of the art Self Supervised NN models that are currently being researched for monocular depth perceptions

2. Related Work

In the single-view depth estimation problem, most works rely on camera motion (Structure-from-Motion), variation in illumination (Shape-from-Shading) or variation in focus. These methods rely on strong assumptions about the scene geometry, hand-crafted features and probabilistic graphical models which exploit horizontal alignment of images or other geometric information.[[9], [10]]. Issues such as lack of scene coverage, scale ambiguities, translucent or reflective materials all contribute to ambiguous cases where geometry cannot be derived from appearance. Another classical approach is to do feature-based matching between a given RGB image and the images of a RGB-D repository in order to find the nearest neighbors; the retrieved depth counterparts are then warped and combined to produce the final depth map.

In practice, the more successful approaches for capturing a scene's depth rely on hardware assistance, e.g. using laser or IR-based sensors, or require a large number of views captured using high quality cameras followed by a long and expensive offline reconstruction process

Recently, methods that rely on CNNs are able to produce reasonable depth maps from a single or couple of RGB input images at real-time speeds. Some of the recent techniques of using neural networks for depth estimation are discussed in the following section.

2.1. Monocular depth estimation

Eigen et al., did the first work to use CNN for depth estimation [2] from monocular images, where the authors used coarse and fine CNN networks to do depth estimation. These ideals were extended to Laina et al [7] who proposed a fully convolutional architecture, encompassing residual learning, to model the ambiguous mapping between monocular images and depth maps. Dan et al [13], experimented with using CNN with Continuous Random Fields (CRF) as a technique, while Hao et al experimented using attention guided networks to further preserve the details of depth maps produced. [4]. While the performance of these methods have been increasing steadily, but quality and resolution of the estimated depth maps have been shown to be improved by other techniques

2.2. Multi-view stereo reconstruction

These CNN algorithms have been recently proposed [6] to solve the problem with more than one image of a scene usually using image pairs [11], or three consecutive frames [3]. All of these methods assume access to the same scene from multiple cameras at the same time.

2.3. Transfer Learning

Transfer learning approaches have been shown to be very helpful in many different contexts. In recent work, Zamir et al. investigated the efficiency of transfer learning between different tasks [14], many of which were related to 3D reconstruction. Ibraheem et. al [1] work on Monocular Depth Estimation via Transfer Learning. The architecture leverages a simple encoder-decoder architecture. The encoder uses an RGB image encoded into a feature vector using the DenseNet-169 network pre-trained on ImageNet.

2.4. Encoder Decoder Methods

Encoder-decoder networks have made significant contributions in many vision related problems such as image segmentation [8], optical flow estimation and image restoration. Ibraheem et. al [1] leverages a simple encoder-decoder architecture. The encoder uses an RGB image encoded into a feature vector using the DenseNet-169 network pretrained on ImageNet. In our project we use Resnet for transfer learning.

Ronneberger et al. [8] presented work on a network and training strategy using data augmentation. Their model architecture consists of downsampling to capture the context and symmetric upsampling for precise localization.

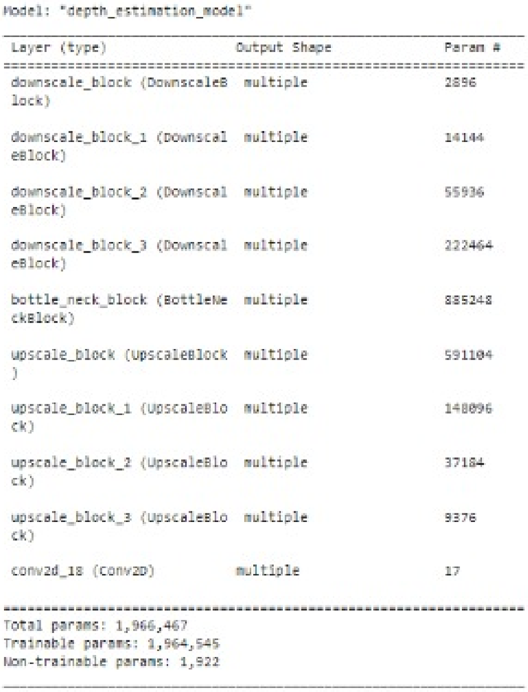
3. Methods

3.1. Network

Our initial baseline model is based on U-Net architecture [8]. Part of the reason for choosing this network as a baseline was its ability to give good results with fewer images.

We built up on the baseline model and added more complexity to the model by adding layers of encoders and decoders as per work of [8]. This network's downscaling block consists of two 3x3 convolutions followed by Leaky Relu activation and batch normalization layer.

We found out that we got better results by using 'Leaky Relu' instead of Relu non linearity activation. We use four of the downsampling blocks each followed by a 2x2 max-pooling operation. At the end we use a bottleneck of two 3x3 convolution layers each followed by LeakyRelu. For the decoder part of the model, the upsampling block has a 2x2 upsampling convolution. It halves the number of feature channels. It is followed by a concatenation with corresponding feature map from downsampling path. Four of these upsampling blocks complete the encoder. at the end a single 3x3 convolution is used to map the feature vectors.



```
Model: "depth_estimation_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 downscale_block (DownscaleB  multiple                 2896
 lock)

 downscale_block_1 (Downscal  multiple                 14144
 eBlock)

 downscale_block_2 (Downscal  multiple                 55936
 eBlock)

 downscale_block_3 (Downscal  multiple                 222464
 eBlock)

 bottle_neck_block (BottleNe  multiple                 885248
 ckBlock)

 upscale_block (UpscaleBlock  multiple                 591104
 )

 upscale_block_1 (UpscaleBlo  multiple                 148096
 ck)

 upscale_block_2 (UpscaleBlo  multiple                 37184
 ck)

 upscale_block_3 (UpscaleBlo  multiple                 9376
 ck)

 conv2d_18 (Conv2D)          multiple                  17

=================================================================
Total params: 1,966,467
Trainable params: 1,964,545
Non-trainable params: 1,922
_________________________________________________________________
```

Figure 1. Baseline Model Graph

After setting up and running the base model we decided to use the transfer learning method and use a pretrained ResNet34 model with weights trained on ImageNet based on the work of Alhashim et al [1]. Transfer learning methods are proven to be very helpful in many different contexts. Zamir et al. investigated the efficiency of transfer learning between different tasks [14] many of which were related to 3D reconstruction. This method is based on idea of transfer learning where we make use of image encoders originally designed for the problem of image classification [[1], [5]]. Authors of this paper found that encoders that do not aggressively downsample the spatial resolution of the input tend to produce sharper depth estimations especially with the presence of skip connections. In this model architecture we employ a single encoder- decoder network which uses previously mentioned ResNet34 model for downsampling the images. We got better results with this method which we will discuss in a later section.

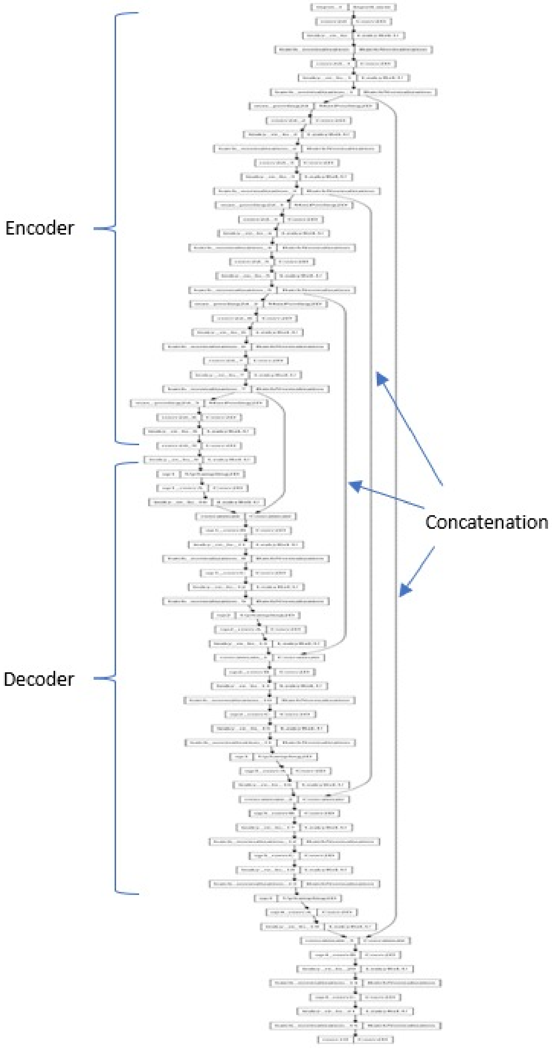


Figure 2. Network architecture of UNet-CNN model

3.2. Dataset

We used the NYU Depth V2 dataset for training purposes. The NYU-Depth V2 data set is composed of video sequences from a variety of indoor scenes as recorded by both the RGB and Depth cameras from the Microsoft Kinect. This dataset provides images and depth maps for different indoor scenes captured at a resolution of 640 × 480. Total dataset contains 120K training images. We used 32K images for our training to accommodate our computing capacity. We split the training data 70% - 30% as training and validation sets. The NYU dataset has 654 test images. We resize the training images to half the original resolution at 320x240. Our network generates predictions at the same input image resolution at 320x240.

3.3. Loss

A standard loss function for depth regression problems considers the difference between the ground truth depth map y and the prediction of the depth regression network $\hat{y}$ [1], [7]. Different loss functions can have a significant effect on the training speed and the overall depth estimation performance. Different variations of loss functions to optimize the performance of neural networks. We used the same loss function as Allhashim et all [1] for this project. Final loss is defined as weighted sum three loss functions as expressed below.

$$L(y, \hat{y}) = \lambda L_{depth}(y, \hat{y}) + L_{grad}(y, \hat{y}) + L_{SSIM}(y, \hat{y}). \tag{1}$$

Let's discuss the three loss terms. The first loss term

L1-loss, or Point-wise depth is the L1 loss is point wise difference between pixels of predicted depth and ground truth depth.

$$L_{depth}(y, \hat{y}) = \frac{1}{n} \sum_{p}^{n} |y_p - \hat{y}_p|. \tag{2}$$

L1 grad loss is L1 loss defined over the image gradient $\boldsymbol{g}$ of the depth image

$$L_{grad}(y, \hat{y}) = \frac{1}{n} \sum_{p}^{n} |\boldsymbol{g}_{\mathbf{x}}(y_p, \hat{y}_p)| + |\boldsymbol{g}_{\mathbf{y}}(y_p, \hat{y}_p)| \tag{3}$$

Structural similarity index(SSIM) loss or L_{SSIM} uses Structural Similarity [9] term. It is often used on image reconstruction methods. SSIM is shown to be a good loss term for depth estimation tasks using CNNs by Godard et al [1], [12] It is defined as

$$L_{SSIM}(y, \hat{y}) = \frac{1 - SSIM(y, \hat{y})}{2}. \tag{4}$$

The total loss is defined as addition of the three losses. Our general understanding is that SSIM loss contributes the most to improving model performance.

We use value of $\lambda = 0.1$ based on the work of Allhashim et al [1].

3.4. Implementation

We used tensorflow libraries and keras api for implementation of both of our networks. As discussed in the network section, our first model is our own implementation of Unet architecture while the second model uses a transfer learning method. In both models we use Adam optimizer. A decaying learning rate is used to speed up learning at initial stages and use a smaller learning rate as we progress. Initial learning rate of 0.0001 is used. Polynomial decay function is used to lower the learning rate as epochs increase. We use a batch size of 16. Total trainable parameters in the unet model are 8.6M. While total trainable parameters in the transfer learning model are 24.8M. We trained the UNet CNN model for 240 iterations. Starting from a learning rate of 0.0001 we end with a learning rate of 0.00006 at the end of training.

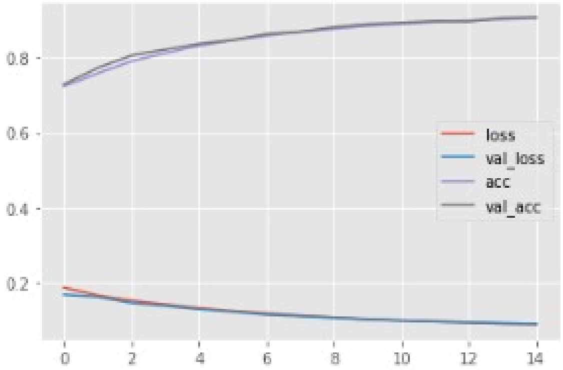


Figure 3. Training, validation: losses and accuracies for UNet based CNN

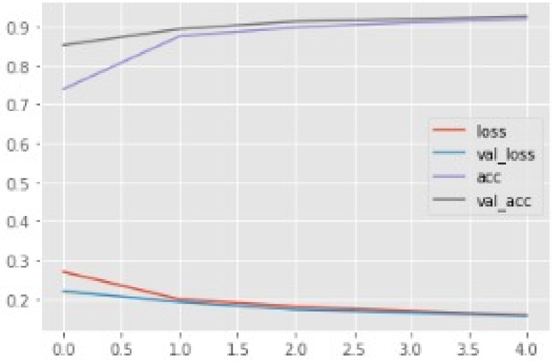


Figure 4. Training, validation: losses and accuracies for transfer learning model

3.5. Results

3.5.1 Quantitative evaluation

We quantitatively compare the results of our models using five metrics mentioned in [1] and based on Eigen et al [2].

Given that y_p is a pixel in depth image y, $\hat{y}_p$ is a pixel in the predicted depth image y while n is the total number of pixels in each image. These parameters are :

- average relative error (rel): $\frac{1}{n}\sum_p^n \frac{|y_p - \hat{y}_p|}{y}$;
- root mean squared error (rms): $\sqrt{\frac{1}{n}\sum_p^n (y_p - \hat{y}_p)^2)}$;
- threshold accuracy (δ_i): % of y_p s.t. $\max(\frac{y_p}{\hat{y}_p}, \frac{\hat{y}_p}{y_p}) = \delta < thr$ for $thr = 1.25, 1.25^2, 1.25^3$;

First and second terms are errors while three terms are accuracies. We want errors to be low and accuracy terms to be higher. Apart from these five above mentioned terms, we also used another custom $L1$ distance based accuracy term to gauge the learning process.

We can see in Table 1, the performance comparison between two of our methods as well as other state of the art models. Our vanilla CNN model did a good job of predicting the depths from given images. Our second model with transfer learning method does better and is somewhat comparable to state of the art networks.

Our models have fewer training parameters and are run for much less number of iterations compared to the models mentioned in the Table 1.

We compared two of our methods in terms of depth accuracy as well. Depth accuracy compares pixel wise distance between predicted depth and true depth. Transfer learning model outperforms the UNet CNN model. A typical source of error for single image depth estimation networks is the estimated absolute scale of the scene [15]. Alhashim et al [1] achieved improvement in their model's performance by multiplying the predicted depths by a scalar that matches the median with the ground truth.

3.5.2 Quantitative evaluation

We had got decent results with the baseline model we used for milestone review. We notice that in the image below Figure 5.

We were able to get better depth maps from both of our models Figure 6 and Figure 7 compared to baseline. Figures 6 and 7 show results of depth estimation maps that are predicted using two of our models along with true depths of the images. Both our approaches produce a good quality of depth estimates. Depth edges on the transfer learning model [Figure 7] are visually better than that of UNet CNN model [Figure 6].

4. Conclusion

In this work we designed and trained network architectures for depth estimation using a single image. We used recent advances in the depth estimation methods and leveraged high performance pretrained models for training purposes. A well constructed model with right weight initialization can give good results. Transfer learning is a very powerful tool and can be used to achieve good results for depth estimation. Study of data augmentation and its effect on the neural network is a major topic for future work. We think data augmentation techniques can certainly be used to improve performance of the model. Improving capture of

Method	δ_1	δ_2	δ_3	rel	rms
Eigen et al. [2]	0.769	0.950	0.988	0.158	0.641
Laina et al. [7]	0.811	0.953	0.988	0.127	0.573
Alhasim et al. [1]	0.895	0.980	0.996	0.103	0.390
Our UNet based CNN	**0.558**	**0.712**	**0.836**	**0.384**	**0.751**
Our Transfer Learning based	**0.661**	**0.784**	**0.837**	**0.292**	**0.682**

Table 1. Quantitative comparison of different methods on NYU V2 dataset.

Method	Depth Accuracy
Our UNet based CNN	0.7584
Our Transfer Learning based	0.8212

Table 2. Depth accuracy comparison on two methods.

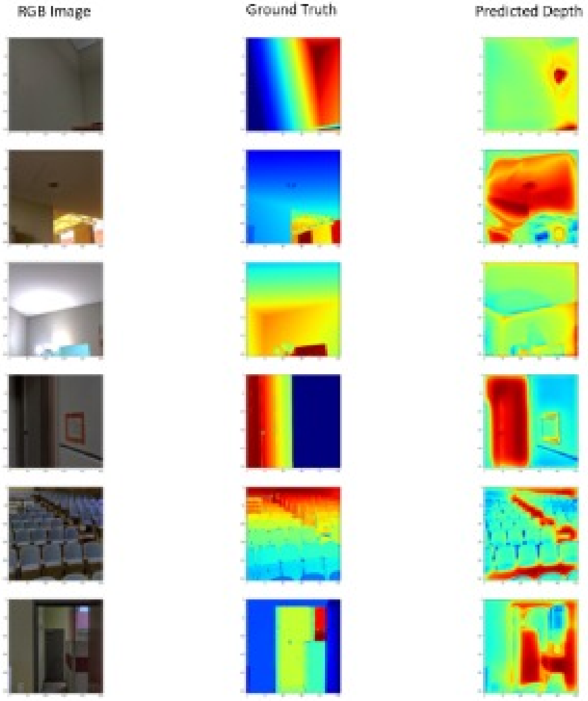


Figure 5. Depth maps results of baseline model

object boundaries is another aspect that can be looked into and improved further.

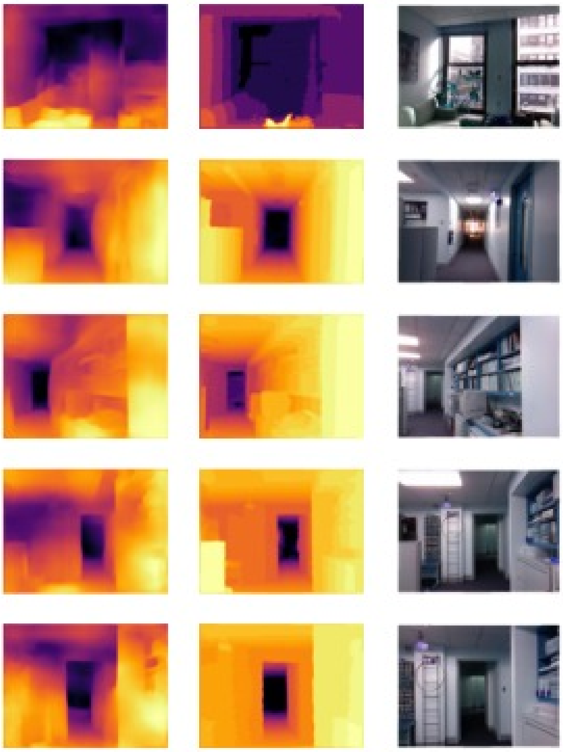


Figure 6. Depth maps results of UNet based CNN model. Starting from left column is the predicted depth map, one in center is true depth and on the right is an original image.

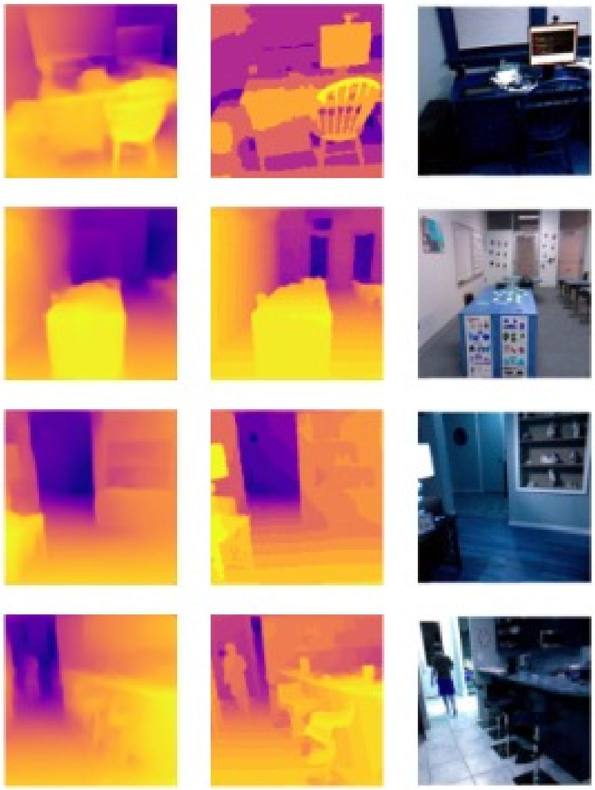


Figure 7. Depth maps results of transfer learning based model. Starting from the left column is the predicted depth map, one in center is true depth and on the right is an original image.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018. 1, 2, 3, 4, 5

[2] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 1, 4, 5

[3] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 2

[4] Zhixiang Hao, Yu Li, Shaodi You, and Feng Lu. Detail preserving depth estimation from a single image using attention guided networks. In *2018 International Conference on 3D Vision (3DV)*, pages 304–313. IEEE, 2018. 2

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2

[6] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018. 2

[7] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016. 1, 3, 5

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 2

[9] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005. 1

[10] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Learning 3-d scene structure from a single still image. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007. 1

[11] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 2

[12] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 3

[13] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5354–5362, 2017. 1

[14] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018. 2

[15] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 4